REIXS Data Storage Strategies: HDF5 and Metadata

Teak D. Boyko *PhD*

Senior Scientist/REIXS Beamline Responsible

Canadian Light Source | Centre canadien de rayonnement synchrotron

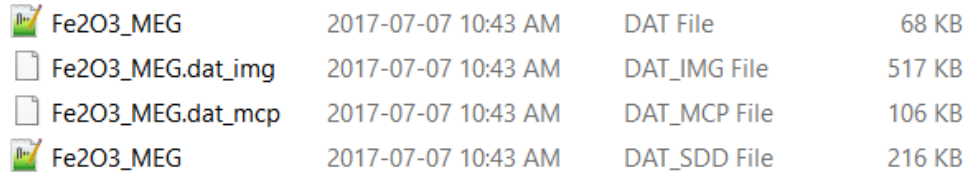
Outline

- How do/did we write experimental data?
  - Standard files and custom auxiliary files.
- Why move to HDF5 and not stick with ASCII files?
  - Data size and organization with higher dimension data sets.
- What is the structure of the HDF5 files?
  - NeXus format and current HDF5 structure.
- What at the advantages of HDF5 (binary files)?
  - Disk space and data access.
- How do users deal with HDF5?
  - Tools available for users.

How did we use to write data?

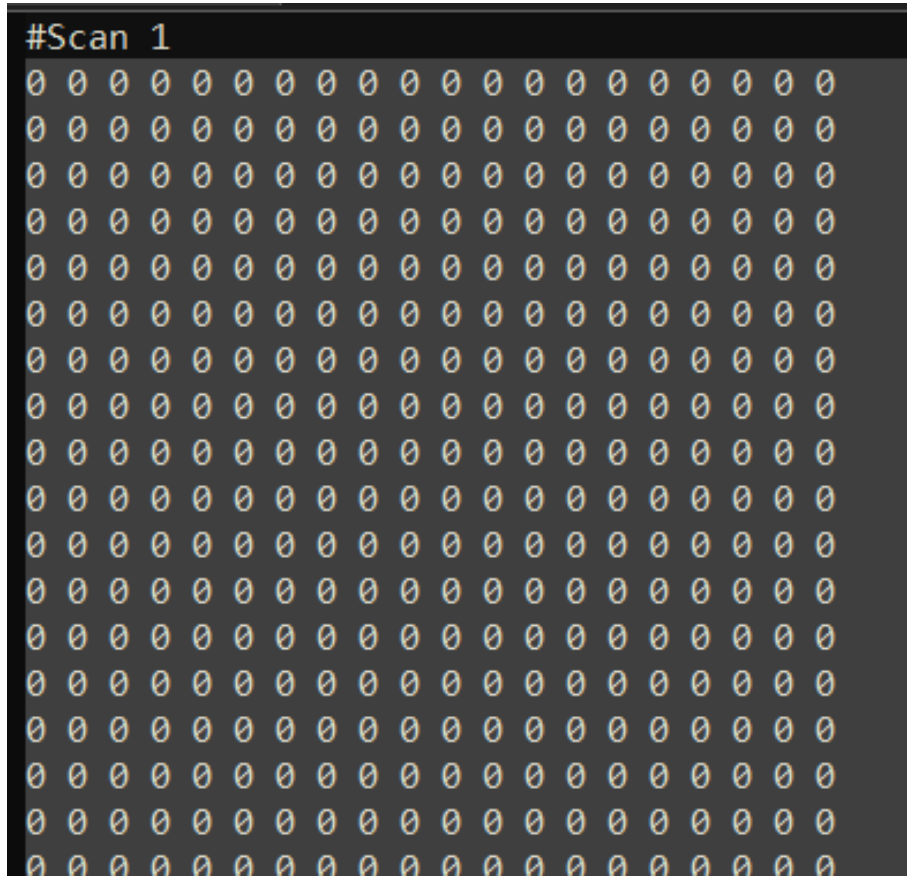
- Prior to 2017 the RIXS ES data acquisition was with in-house Software *Acquaman*
  - Binary format: CDF
    - Export final data in ASCII
  - Phased out during 2016-2017 upgrade
  - RSXS ES always used FOURC
    - Only wrote to one ASCII file.
- REIXS started using SPEC/FOURC for all data acquisition in 2017
  - Main SPEC/FOURC file: ASCII
    - SCAs only
  - Auxiliary files added for RIXS (later RSXS): ASCII
    - SDD needs MCAs
    - XES Spectrometer needs both MCAs and images.

```
#F N_calib.dat
#E 1674573277
#D Tue Jan 24 09:14:37 2023
#C RIXS  User = boykocls

#O0 Samp X  Samp Y  Samp Z  Samp Th  Samp Horz  Samp Depth  Samp Vert  Samp Angle
#O1 Detect Z  Spectr Rot  MCP A Tilt  MCP B Tilt  Spectr Tran  XES Angle  XES Dist  MCP Angle
#O2 ES Trans  Energy  LI_Angle  Hex X  Hex Y  Hex Z  Hex U  Hex V
#O3 Hex W
#o0 ssx ssy ssz ssth ssh ssd ssv ssa
#o1 detz spr deta detb spt spa spd dta
#o2 est engy lian hex_x hex_y hex_z hex_u hex_v
#o3 hex_w
#J0 Dwell  Mesh  Sample  MCP_A  MCP_B  SDD_Tot  SDD_ROI  CEM
#J1 XEOL_Max  Mesh_Curr  Samp_Curr  Mono_Engy  Ring_Curr  Samp_Temp  Test_Engy
#j0 sec i0 tey mcpa mcpb sddt sddr cem
#j1 xeol i0_a tey_a beam ring temp testE


#S 1  ascan  engy 380.005 380.005  2 60
#D Tue Jan 24 09:16:15 2023
#T 60  (Dwell)
#G0 0
#G1 0
#G3 0
#G4 0
#Q
#P0 -4.8337997 -0.67100037 124.7 290.625 0 0 -2 62.5
#P1 18.75 200.5044 17.254718 12.501289 952.9702 7.5135 952.15 5.454
#P2 -1.9444649 379.995 0 18.805 2.111 17.963 0.021 0
#P3 0
#N 17
```



How did we use to write data?

- Prior to 2017 the RIXS ES data acquisition was with in-house Software Acquaman
  - Binary format: CDF
    - Export final data in ASCII
  - Phased out due to lack of support
  - RSXS ES always used FOURC
    - Only wrote to one ASCII file.
- REIXS started using SPEC/FOURC for all data acquisition in 2017
  - Main SPEC/FOURC file: ASCII
    - SCAs only
  - Auxiliary files added for RIXS (later RSXS): ASCII
    - SDD needs MCAs
    - XES Spectrometer needs both MCAs and images.



Fe2O3_MEG	2017-07-07 10:43 AM	DAT File	68 KB
Fe2O3_MEG.dat_img	2017-07-07 10:43 AM	DAT_IMG File	517 KB
Fe2O3_MEG.dat_mcp	2017-07-07 10:43 AM	DAT_MCP File	106 KB
Fe2O3_MEG	2017-07-07 10:43 AM	DAT_SDD File	216 KB

How do we currently write data?

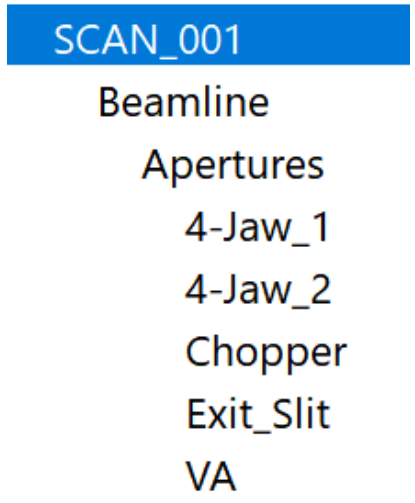
- Major Data Overall in 2018/2019
  - Consistent format between RSXS ES and RIXS ES
    - SPEC/FOURC headers in auxiliary files.
  - Higher level data orginzation
    - Image stacks: RSXS only
    - Detector scales added
  - Data organized by date and group
  - Secure access with group user accounts
- Development of HDF5 in 2020
  - Introduced on RIXS ES in late 2022
    - Now the standard for data access.
  - Introduced on RSXS ES in early 2023
    - Users still reliant on ASCII files.

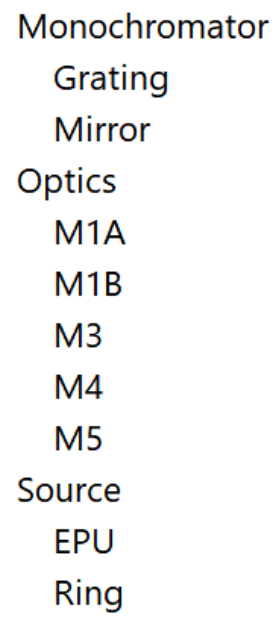
```
#S 1 rscan engy 845 875 20 1
#D Sat Dec 18 16:04:46 2021
#N 1024
#T 1
#C MCP Energy Scale
820.860107
820.941223
821.0224
```

```
#S 1 rscan engy 845 875 20 1
#D Sat Dec 18 16:04:46 2021
#N 1024
#T 1  (Dwell)
#C SDD Energy Scale
#@CALIB -20.019 2.6698
-20.0189991
-17.3491993
-14.6794004
```

Where do we go from here? HDF5

- More and more files..
  - Now 3 MCA detectors (SDD, XES, XEOL): but adding two more
    - Would need 2 additional files.
    - What about data retention and organization?
- HDF5: Organized and compact
  - Multiple scans per file
  - Each detector can be grouped and aliased in standard locations
    - Higher dimensional data is easily appended
    - Stacks, images, lines
  - Extremely rich metadata
    - Beamline snapshot captured before each scan





Dataset	Rank	Dim Sizes	Type	Value
Q1	1	1;	32-bit LE float	37.498
Q2	1	1;	32-bit LE float	0
Q3	1	1;	32-bit LE float	-37.498
Q4	1	1;	32-bit LE float	0
gap	1	11;	64-bit LE float	214.721, 214.721,
harmonic	1	1;	32-bit signed	1
lian	1	1;	64-bit LE float	0
offset	1	11;	64-bit LE float	182.061, 182.061,
polarization	1	1;	string	"LINEAR VERT -"

What is NeXus and should we follow the standards?

- NeXus is an international standard for HDF5 files.
  - http://www.nexusformat.org
  - Dictates required attributes and organization of the HDF5 file.
- REIXS will make efforts to adhere to NeXus standards.
  - Ensure future consistency.

3.3.2.26. NXxas

Status:

application definition, extends NXobject

Description:

This is an application definition for raw data from an X-ray absorption spectroscopy experiment.

This is essentially a scan on energy versus incoming/ absorbed beam.

SCAN_XXX

Beamline

- Apertures
- Diagnostics
- Monochromator
- Optics
- Source

Data

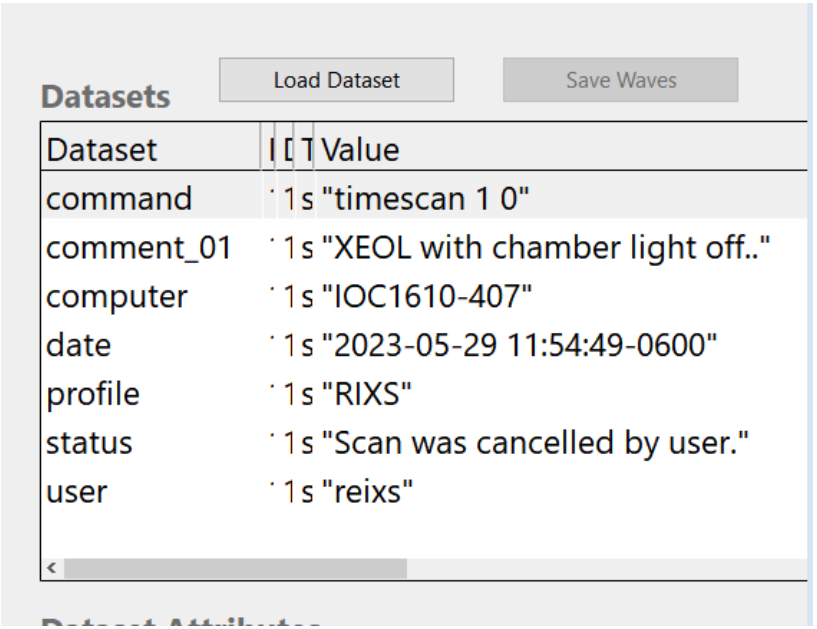
Endstation

- Counters
- Detectors
- Motors
- Sample
- Translation
- Vacuum

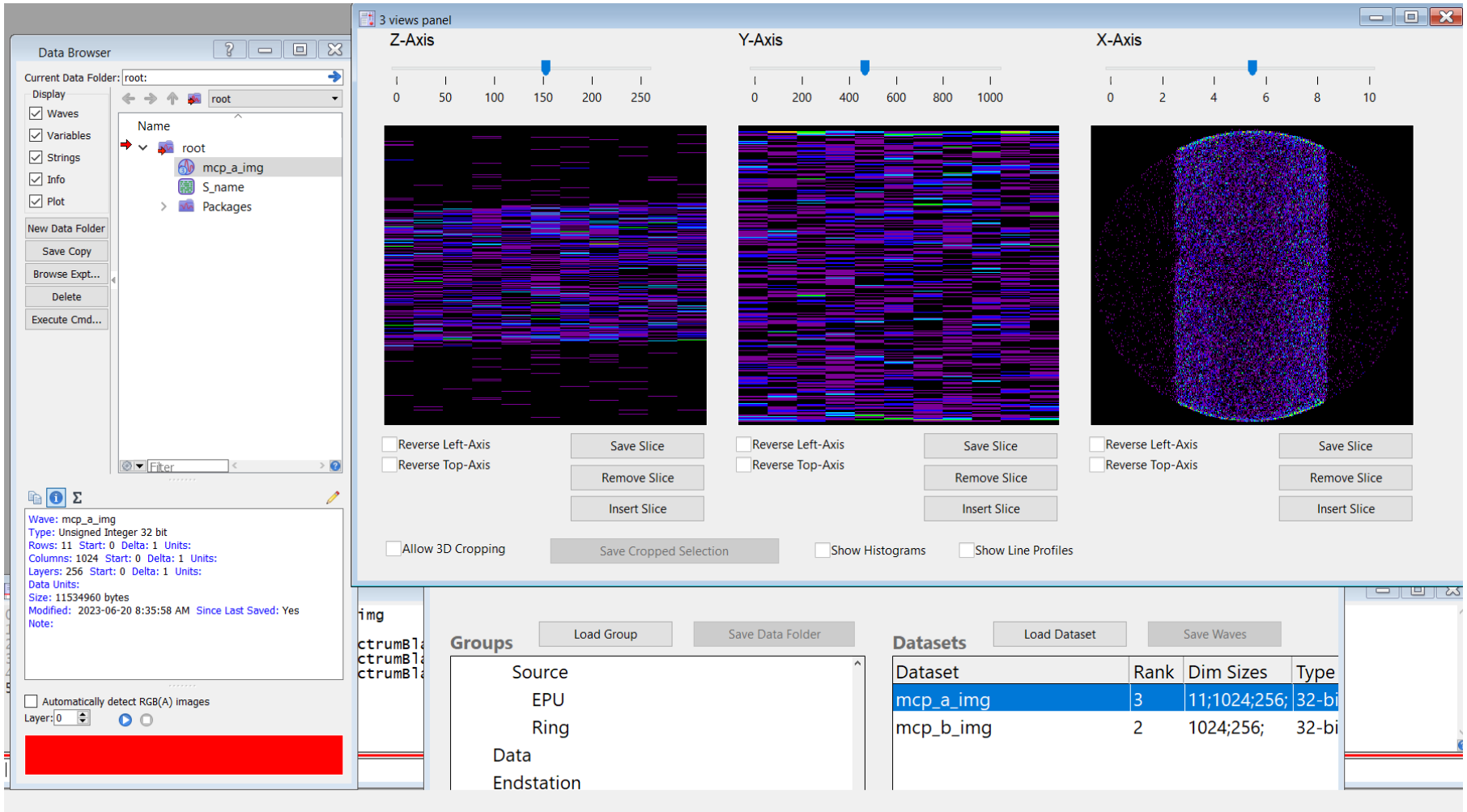
Why not stick with ASCII files? Why HDF5?

- ASCII files require the entire file to be parsed.
  - Large ASCII files need to be loaded into memory.
  - Access time scales with file size.
- ASCII files can not be modified unsequentially.
  - New data can not be easily inserted, only appended to the end.
  - HDF5 does not support duplicate data, but more data can always be added.
- HDF5 supports linking.
  - Redundant data can be well organized without additional overhead.
- Working with data is easier.
  - Large images and stacks can directly loaded into arrays for data reduction.

- Comments can be inserted retroactively after the scans have finished.
  - Overnight macro.



Why not stick with ASCII files? Why HDF5?



Canadian Light Source
Centre canadien de rayonnement synchrotron

Is HDF5 more efficient for data storage?

- The short answer: sometimes, but mostly no.
  - Increase meta data.
  - More complex data sets.
- We just end up making more data because it is easier.
- HDF5 can store data more efficiently if it is not mostly zeros..
  - Space in ASCII for zero in array (2 bytes)
  - 32 bit/64 bit unsigned in HDF5 (4-8 bytes)
  - Space saved if the average value is 5-6 digits (6-7 bytes)
- Large files create delays in network synchronization.
  - Working solution for passive synchronization.
  - Deploy in next cycle.

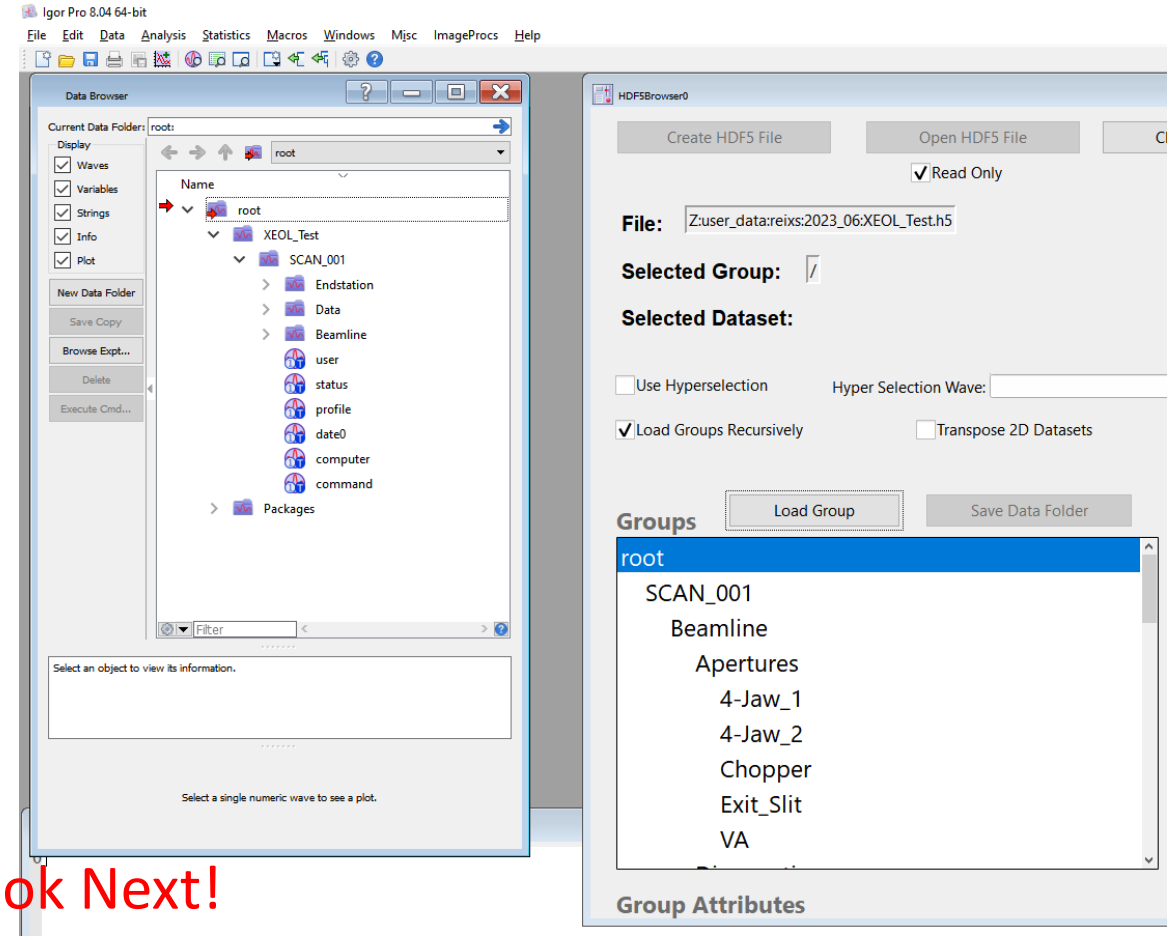
- Gigabit connect for syncing
  - 125MB/s or 8s for 1GB files

Type		Size
H5 File	108	883,441 KB
H5 File	61	646,085 KB
H5 File	41	336,799 KB
H5 File	34	265,228 KB
H5 File	28	217,804 KB

≈10 MB/scan

How to interface with HDF5?

- Many options for HDF5
  - Igor (built-in browswer)
  - Linux (h5dump)
  - Most programming languages have a hdf5 library
  - HDF group HDF VIEW
    - https://www.hdfgroup.org/downloads/hdfview/
  - Beamline supported Jupyter Notebook
    - https://pypi.org/project/reixs/



Patrick Braun will talk about our Jupyter Notebook Next!

Outlook - Questions for you? Or for me…

- Phasing out dates for ASCII
  - RIXS ES: **October 2023**
    - ASCII files are already not being used, just there as a backup.
  - RSXS ES: **January 2024**
  - Only main SPEC/FOURC data file will be available and main HDF5 file.

- Are there concerns with the phasing out of ASCII data?

- What metadata would you like that isn't there?